



人事管理系统

Personnel Management System (PMS)

软件需求规格[SRS]文档

Version: 1.0.1

2024-3-25

All Rights Reserved

教师：余仲星

队员：马豪 李晴 高潼 全鑫 耿尊至

目录

软件需求规格说明 (SRS).....	1
1 范围.....	4
1.1 标识.....	4
1.2 系统概述.....	4
1.3 文档概述.....	5
1.4 基线.....	6
2 引用文件.....	6
3 需求.....	7
3.1 所需的状态和方式.....	7
3.2 需求概述.....	9
3.2.1 目标.....	9
3.2.2 运行环境.....	13
3.2.3 用户的特点.....	14
3.2.4 关键点.....	15
3.2.5 约束条件.....	16
3.3 需求规格.....	18
3.3.1 软件系统总体功能/对象结构.....	18
3.3.2 软件子系统功能/对象结构.....	18
3.3.3 描述约定.....	18
3.4 CSCI 能力需求.....	19
3.5 CSCI 外部接口需求.....	20
3.5.1 接口标识和接口图.....	错误!未定义书签。
3.6 CSCI 内部接口需求.....	21
3.7 CSCI 内部数据需求.....	22
3.8 适应性需求.....	23
3.9 保密性需求.....	24
3.10 保密性和私密性需求.....	25
3.11 CSCI 环境需求.....	25
3.12 计算机资源需求.....	26
3.12.1 计算机硬件需求.....	26
3.12.2 计算机硬件资源利用需求.....	26
3.12.3 计算机软件需求.....	26
3.12.4 计算机通信需求.....	26
3.13 软件质量因素.....	27
3.14 设计和实现的约束.....	32
3.15 数据.....	32
3.16 操作.....	34
3.17 故障处理.....	35
3.18 算法说明.....	36
3.19 有关人员需求.....	42
3.20 有关培训需求.....	43
3.21 有关后勤需求.....	44

3.22 其他需求	44
3.23 包装需求	46
3.24 需求的优先次序和关键程度	46
4 合格性规定	46
5 需求可追踪性	48
6 尚未解决的问题	49
7 注解	49
附录	50

1 范围

1.1 标识

本条应包含本文档适用的系统和软件的完整标识，(若适用)包括标识号、标题、缩略词语、版本号和发行号。

本文档适用于以下系统和软件：

- 操作系统：Windows 10（版本号：21H2）
- 集成开发环境：Pycharm（版本号：2021.1.1）
- 集成开发环境：Vscode（版本号：2022.12.1）
- 开发语言：python（版本号：3.9.0）
- 标记语言：Markdown

缩略词语：

- Windows 10：Win10
- Vscode：VS
- Python：py

本文档适用于 Win10 操作系统，使用 Pycharm 和 vscode 集成开发环境和 py 开发语言（版本号为 3.9.0），采用 markdown 和 python 语言开发一个人事管理系统。

1.2 系统概述

本系统是一个针对人事管理的综合性软件平台，旨在帮助组织高效管理人力资源，提升管理效率和员工满意度。以下是本系统的一般特性、历史、相关方和运行现场的概述：

系统特性：

提供员工信息管理：包括基本信息、合同信息、薪资信息等，实现信息的集中化管理和自动化处理。

自动化流程支持：支持入职、离职、转正等人事流程的自动化执行，减少人工干预，提高流程效率。

数据分析和报表功能：提供各种定制化的报表和数据分析功能，帮助管理层进行决策分析。

安全可靠：采取多层次的安全措施保护员工信息，符合相关法规和政策要求。

用户友好界面：设计简洁直观的界面，提升用户体验，降低操作门槛。

历史:

本系统的开发源于对传统人事管理方法的不足和现有系统无法满足需求的认识。历经多次需求调研和技术评估，经过团队的努力开发，不断迭代更新，逐步形成了今天的完整功能。

相关方:

投资方：期待。

需方：人力资源部门作为主要需方，提出需求和参与系统的使用和测试。

用户：人事管理系统公司所有员工都是系统的最终用户，他们将使用系统进行个人信息维护和流程操作。

开发方：山东大学**开发团队。

支持机构：系统运行过程中可能需要技术支持的第三方服务提供商。

运行现场:

当前运行现场：本系统将在各种公司内部部署，作为内部人力资源管理的主要工具，在公司的办公场所使用。

计划的运行现场：未来可能会考虑将系统部署到云端，提供在线服务，以适应远程办公和分布式团队的需求。

其他相关文档:

(1) 开发需求文档：这个文档主要描述人事管理系统 PMS 的开发需求，包括应用场景、目标用户、功能点、界面设计、技术选型、业务流程以及其他相关需求。

(2) UI 设计文档：这个文档包含人事管理系统 PMS 的交互设计、视觉设计和规范，其中包括设计元素、颜色、图标、字体等。

(3) 产品规范文档：这个文档描述了人事管理系统 PMS 的标准、规范和要求，包括应用程序界面、着装、界面模板、版权等。

(4) 数据库设计文档：这个文档描述了人事管理系统 PMS 的数据库模型和规范，包括数据表、关系和 SQL 语句等。

(5) 测试文档：这个文档描述了人事管理系统 PMS 的测试方法和系列测试任务流程，包括各个模块、各个功能测试过程及其结果。

(6) 部署文档：这个文档描述了人事管理系统 PMS 的部署和维护指导，包括运行环境、应用程序版本控制、应用程序部署、日志记录、错误处理等。

(7) 用户手册：这个文档容纳了人事管理系统 PMS 的操作说明书，包括功能介绍、使用方式、注意事项、用户操作流程等。该文档一般是面向 app 的最终用户。

(8) 技术文档：这个文档描述了人事管理系统 PMS 的实现细节、编码规范、架构设计、测试方法、部署说明、维护细节等，多是面向人事管理系统 PMS 的开发人员。

1.3 文档概述

本文档所涵盖的是软件需求规格说明（SRS），目的是为开发团队、客户以及其他相关人员提供详细的系统开发需求规格说明。文档内容包括了系统需求之一切方面，如功能性需求、

性能需求、接口需求、设计约束、质量属性，以及其他非功能性需求等。此外，文档还对系统输入输出、用户界面设计、数据管理和故障处理等方面的需求做了详细描述。

在文档的使用过程中，需注重其保密性或私密性要求，未授权人员不得浏览或使用文档中的信息。对文档的更新和修改必须进行记录及管理，以确保其准确性和一致性。为保证项目开发的质量，软件需求文档是非常重要的，它能够有效地指导系统的设计和实现，并为项目开发的进展提供可靠依据。因此，我们应该重视软件需求文档的编写和管理工作，以保证项目开发的成功。

1.4 基线

本系统需求规格说明书所参考的基线：在系统的设计过程中，基线是一个既定的标准或版本，用于参考和比较。这些标准或版本可包括系统的硬件规格、软件设计要求、系统体系结构等，是系统设计中必不可少的依据。在编写本系统设计说明书前，已经完成了需求分析、系统设计以及软件开发等一系列工作，并已建立了相应的基线。本文档所依据的基线包括以下内容：

1. 需求规格说明书的最终版本；
2. 系统架构设计的最终版本；
3. 硬件规格的最终版本；
4. 软件设计规范的最终版本；
5. 集成测试计划的最终版本；
6. 配置管理计划的最终版本；

在编写过程中，我们将遵循以上基线，并视实际情况进行必要的修改和完善。同时，本文档的编写也应该遵守代码规范和标准，保证文档质量和系统可靠性。基线的建立和维护对于整个系统开发过程至关重要，有助于确保系统设计的一致性和稳定性。因此，我们应该认真对待系统基线的编写和管理工作，以确保项目开发的顺利进行。

2 引用文件

本章应列出本文档引用的所有文档的编号、标题、修订版本和发行日期，也应标识不能通过正常的供货渠道获得的所有文档的来源。

1. 开源软件协议
 - (1) 编号： 1
 - (2) 开源软件协议修订版本： 1.0
 - (3) 发行日期： 2023.04.29
 - (4) 来源：可在开源软件官网或者相关网站进行查看和下载
2. 开发工具文档
 - (1) 编号： 2
 - (2) 开发工具文档修订版本： 1.1
 - (3) 发行日期： 2023.03.29
 - (4) 来源：可在开发工具官网或者相关网站进行查看和下载
3. 软件测试文档
 - (1) 编号： 3

- (2) 软件测试文档版本: 2.0
- (3) 发行日期: 2022.04.29
- (4) 来源: 可在软件测试机构或者相关网站进行查看和下载
- 4. 系统设计文档
 - (1) 编号: 4
 - (2) 系统设计文档修订版本: 1.2
 - (3) 发行日期: 2022.01.19
 - (4) 来源: 由软件架构师和系统设计专家编写
- 5. 法律法规指南
 - (1) 编号: 5
 - (2) 法律法规指南修订版本: 2.3
 - (3) 发行日期: 2024.01.19
 - (4) 来源: 由法律顾问或律师事务所编写, 提供关于劳动法、数据隐私等其他方面的解读和指导。

3 需求

本章应分以下几条描述 CSCI 需求, 也就是, 构成 CSCI 验收条件的 CSCI 的特性。CSCI 需求是为了满足分配给该 CSCI 的系统需求所形成的软件需求。给每个需求指定项目唯一标识符以支持测试和可追踪性。并以一种可以定义客观测试的方式来陈述需求。如果每个需求有关的合格性方法(见第 4 章)和对系统(若适用, 子系统)需求的可追踪性(见 5.a 条)在相应的章中没有提供, 则在此进行注解。描述的详细程度遵循以下规则: 应包含构成 CSCI 验收条件的那些 CSCI 特性, 需方愿意推迟到设计时留给开发方说明的那些特性。如果在给定条中没有需求的话, 本条应如实陈述。如果某个需求在多条中出现, 可以只陈述一次而在其他条直接引用。

3.1 所需的状态和方式

如果需要 CSCI 在多种状态和方式下运行, 且不同状态和方式具有不同的需求的话, 则要标识和定义每一状态和方式, 状态和方式的例子包括: 空闲、准备就绪、活动、事后分析、培训、降级、紧急情况和后备等。状态和方式的区别是任意的, 可以仅用状态描述 CSCI, 也可以仅用方式、方式中的状态、状态中的方式或其他有效方式描述。如果不需要多个状态和方式, 不需人为加以区分, 应如实陈述; 如果需要多个状态或方式, 还应使本规格说明中的每个需求或每组需求与这些状态和方式相关联, 关联可在本条或本条引用的附录中用表格或其他的方法表示, 也可在需求出现的地方加以注解。

3.1.1 CSCI 在交付给用户之前的运行需求:

- (1) 系统必须经过一系列的单元测试、集成测试和系统测试, 以确保满足全部功能需求和性能要求。

- (2) 系统必须具有高可靠性、可用性和可维护性，能够保证其在用户操作下不会崩溃并且能够快速恢复。
- (3) 系统必须完全符合软件架构设计和软件规范要求，方便地配置集成到整个系统中。
- (4) 系统必须有详尽的使用手册和技术文档，使用户能够准确了解其使用方法和维护方法。
- (5) 系统必须具备良好的可扩展性和升级性，以便系统的后续开发和升级。

3.1.2 CSCI 在正常运行状态下的运行需求：

- (1) 系统必须具有高性能和可靠性，能够快速响应用户请求并保证数据的完整性。
- (2) 系统必须符合所有用户需求和系统架构设计的要求，能够与其他组件无缝协作。
- (3) 系统必须有良好的用户界面和交互体验，符合用户习惯和经验。
- (4) 系统必须具有稳定的运行环境、可维护性和可扩展性，能够随时响应用户的需求和系统的变化。

3.1.3 CSCI 在异常和故障状态下的运行需求：

- (1) 系统必须具备错误处理机制，能够及时捕获并给出相应的提示和解决方法。
- (2) 系统必须具有良好的日志记录机制，记录所有的运行日志和异常日志，并方便查看和管理。
- (3) 系统必须具有快速恢复能力，当发生故障时能够及时恢复系统并保证数据不会丢失。
- (4) 系统必须具备安全保护措施，保障系统的安全性与合规性。

3.1.4 CSCI 在用户操作下的运行需求：

- (1) 系统必须能够在用户操作下稳定运行，并对用户的操作进行及时响应。
- (2) 系统必须具备良好的用户界面设计，简洁明了、易于操作，降低用户的学习和使用成本。
- (3) 系统必须能够有效地处理用户输入，并根据用户的需求提供相应的反馈和结果。
- (4) 系统必须确保用户操作的安全性和可靠性，避免因用户操作而引发的系统崩溃或数据损坏。

3.1.5 CSCI 在多用户并发访问下的运行需求：

- (1) 系统必须能够有效地处理多用户同时访问的情况，保证系统的稳定性和性能。
- (2) 系统必须具备合适的并发访问控制机制，确保各用户之间的数据安全性和隔离性。
- (3) 系统必须能够根据用户的访问需求进行资源分配和调度，避免资源争用和性能下降。
- (4) 系统必须具备合理的负载均衡机制，平衡各个用户请求的处理负荷，确保系统整体性能稳定性。

3.2 需求概述

3.2.1 目标

- a. 本系统的开发意图、应用目标及作用范围(现有产品存在的问题和建议产品所要解决的问题)。
- b. 本系统的主要功能、处理流程、数据流程及简要说明。
- c. 表示外部接口和数据流的系统高层次图。说明本系统与其他相关产品的关系,是独立产品还是一个较大产品的组成部分(可用方框图说明)。

A. 痛点概述:

随着组织规模的扩大和管理需求的增加,传统的人事管理方法已经无法满足日益增长的需求。主要痛点包括:

手工管理效率低下: 随着员工数量的增加,手工管理变得耗时且容易出错,导致人事信息管理的混乱和不可控。现有的手工方法已经无法有效应对组织规模的扩大。

现有系统功能不足: 组织可能已投入相当资源使用现有系统,但随着管理需求的变化,现有系统可能无法满足新的管理需求。功能可能过时,界面设计不友好,操作流程繁琐,降低了用户的工作效率和满意度。

繁琐的手工操作: 传统的人事管理方法通常需要大量的手工操作,包括数据录入、信息更新和报表生成等。这些手工操作不仅耗时、耗力,而且容易出错,降低了管理效率和员工满意度。

法规要求的变化: 随着法规要求的不断变化,现有系统可能不符合新的法规要求,可能面临法律风险和罚款。因此,需要更新系统以满足新的法规要求,确保数据安全和合规性。

管理决策需要更准确的数据支持: 管理层需要更准确、更科学的数据支持管理决策。他们希望有一个更强大的人事管理系统,能够提供全面的、实时的数据分析和对员工信息的深度挖掘和预测功能,以支持组织的战略规划和业务决策。

B. 系统目标:

基于上述痛点,人事管理系统(PMS)的主要目标包括:

提高管理效率和准确性: 实现自动化、集中化的员工信息管理,能够快速、准确地进行数据录入、修改和查询,从而提高管理效率和准确性。

优化用户体验: 提供简洁、直观的用户界面,减少操作复杂度,提升用户体验,增加用户满意度。

智能数据处理和分析： 使用智能算法和数据分析技术对大量的人事数据进行处理和分析，挖掘有用的信息，帮助管理人员做出更明智的决策，提升管理水平和效率。

保障数据安全和合规性： 更新系统以满足新的法规要求，加强数据隐私和安全性措施，确保员工个人信息的安全和隐私保护，避免法律风险。

支持系统的扩展和定制： 设计系统具备良好的扩展性和定制性，能够适应组织未来的发展和变化，满足不同用户的个性化需求。

B. 本系统的主要功能、处理流程、数据流程及简要说明

1. 主要功能概述：

员工信息管理：

- 基本信息录入： 系统提供了用户友好的界面，包括表单和字段验证，确保录入的员工基本信息完整和准确。

- 学历信息录入： 用户可以逐步录入员工的学历信息，系统支持多种学历类型和学校名称的输入，并提供自动完成功能以便快速选择。

婚姻状况录入： 用户可以选择员工的婚姻状况，系统根据选择的状态显示相应的字段，如配偶姓名等。

- 其他相关信息录入： 系统提供了灵活的字段设置功能，允许管理员根据需要添加或删除员工的其他相关信息字段，确保系统的扩展性和定制性。

信息修改：

- 实时更新： 用户可以随时修改员工的各种信息，系统提供了即时更新功能，确保修改后的信息立即生效。

修改记录追溯： 系统记录了每次信息修改的时间、操作人员和修改内容，管理员可以查看修改记录并追溯信息的变更历史。

员工状态管理：

- 状态标记功能： 系统提供了简单直观的操作界面，让管理员可以轻松地对员工的状态进行标记，如转出、辞职、退休等。

状态展示： 在员工信息页面，系统以明显的方式展示员工的当前状态，方便管理员快速了解员工的工作状态。

工资管理：

- 工资项目配置： 管理员可以通过系统界面灵活地配置工资项目，包括工资、奖金、扣款等，根据组织的具体需求进行定制。

工资条生成： 系统根据员工的工资项目和规则自动生成工资条，确保工资计算的准确性和

一致性。

- 工资条展示： 员工可以通过系统登录账户查看个人的工资条，系统以清晰的方式展示工资明细和总额，方便员工核对和查看。

查询与统计：

- 多条件查询： 系统提供了灵活的查询功能，用户可以根据员工姓名、部门、职位等多种条件进行快速查询，确保准确获取所需信息。

统计分析： 系统实现了对员工信息的统计分析功能，管理员可以根据学历、婚姻状况、岗位等方面进行信息汇总和分析，生成报表或图表，帮助管理人员深入了解员工情况。

系统安全性：

- 用户身份验证： 系统提供了安全的登录功能，用户需要输入用户名和密码进行身份验证，确保只有授权用户能够访问系统。

访问控制： 系统根据用户角色和权限设置，控制用户对不同功能模块和数据的访问权限，保障数据的安全性和机密性。

数据加密： 对敏感信息进行加密存储和传输，使用加密算法确保数据的安全性，防止数据泄露和非法访问。

2. 处理流程：

员工信息管理：

基本信息录入：

用户打开员工信息管理界面。

填写员工的基本信息，包括姓名、性别、出生日期、联系方式等。

点击保存按钮，系统进行表单验证，验证通过后保存员工基本信息。

学历信息录入：

在员工信息界面，点击添加学历信息按钮。

逐步填写学历信息，包括学历类型、学校名称、专业、入学时间、毕业时间等。

点击保存按钮，系统保存学历信息并自动展示在员工信息页面。

婚姻状况录入：

在员工信息界面，选择员工的婚姻状况，如已婚、未婚等。

根据选择的状态显示相应的字段，如配偶姓名、结婚日期等。

点击保存按钮，系统保存婚姻状况信息。

其他相关信息录入：

管理员根据需要添加或删除员工的其他相关信息字段。

在员工信息界面，点击添加字段按钮，填写字段名称和值。

点击保存按钮，系统保存其他相关信息。

信息修改：

用户进入员工信息界面，点击编辑按钮。

修改员工信息，如联系方式、家庭住址等。

点击保存按钮，系统更新员工信息并记录修改历史。

员工状态管理：

在员工信息界面，管理员点击标记状态按钮。

选择员工的状态，如转出、辞职、退休等。

系统更新员工状态并在界面上展示。

工资管理：

管理员进入工资管理界面，配置工资项目和规则。

系统根据规则自动生成工资条。

员工登录系统查看个人工资条。

查询与统计：

多条件查询：

用户进入员工信息查询界面。

根据姓名、部门、职位等条件进行查询。

系统返回符合条件的员工信息列表。

统计分析：

管理员进入统计分析界面。

根据学历、婚姻状况、岗位等方面进行信息汇总和分析。

系统生成报表或图表，帮助管理人员深入了解员工情况。

3. 数据流程：

员工信息管理：

员工基本信息录入：

用户输入员工的基本信息。

系统将信息保存到员工信息数据库中。

学历信息录入：

用户输入员工的学历信息。

系统将学历信息保存到员工信息数据库中。

婚姻状况录入：

用户选择员工的婚姻状况。

系统将婚姻状况信息保存到员工信息数据库中。

其他相关信息录入：

用户输入员工的其他相关信息。

系统将其他相关信息保存到员工信息数据库中。

信息修改：

用户修改员工的信息。

系统更新员工信息并记录修改历史。

员工状态管理：

管理员标记员工的状态。

系统更新员工状态并记录状态变更历史。

工资管理：

管理员配置工资项目和规则。

系统根据规则自动生成工资条并保存到工资信息数据库中。

查询与统计：

多条件查询：

用户输入查询条件。

系统根据条件在员工信息数据库中检索符合条件的员工信息。

返回查询结果给用户。

统计分析：

管理员选择统计分析的维度和指标。

系统从员工信息数据库中提取数据，进行统计分析。

生成报表或图表展示分析结果。

C. 表示外部接口和数据流的系统高层次图。说明本系统与其他相关产品的关系，是独立产品还是一个较大产品的组成部分：

表 1 主要外部接口表

接口	功能
OAuth/OpenID Connect 服务	用于实现单点登录功能，让员工可以使用他们的社交媒体账号登录系统。
支付宝/微信支付 SDK	用于实现工资发放功能，员工可以通过系统收到工资。
腾讯云短信 API	通过该接口向用户发送验证码
QQ 邮箱邮件 API	通过该接口向用户发送邮件
LDAP/AD 服务	用于集成公司现有的身份验证系统，方便员工使用统一的账号和密码登录系统。
百度智能云图像识别 API	识别员工上传的证件照片，例如身份证、员工证等，用于员工身份认证和信息录入。

3.2.2 运行环境

硬件环境：

设备：支持智能手机或平板电脑。

操作系统：要求设备搭载 win10 或以上版本。

处理器：至少要求四核处理器，以保证系统运行的流畅性。

内存：系统要求至少具备 2GB RAM，以确保应用程序的正常运行。

存储空间：至少需要 500MB 的空闲存储空间，用于安装和存储系统数据。

支持环境：

云存储服务：开发者可能会使用云存储服务如 AWS S3 或阿里云 OSS，以存储系统中的员工信息和相关数据。

数据库：系统会使用关系型数据库如 MySQL，用于存储员工信息、工资记录等数据。

加密库：系统可能会使用加密库来加密用户敏感信息，确保数据的安全性。

第三方 API：系统集成第三方 API，如支付宝/微信支付 SDK，用于实现工资发放功能。

日志记录工具：系统可能会使用日志记录工具来记录用户操作和系统运行状态，以便进行故障排查和系统优化。

安全性验证服务：系统可能会使用身份验证服务如 LDAP/AD 服务，用于用户登录认证和权限

管理。

3.2.3 用户的特点

管理员：

系统配置和管理：管理员拥有对系统各项功能进行配置和管理的权限，包括添加、编辑、删除用户账号，设置用户权限，配置系统参数等。

数据管理和维护：管理员负责管理系统中的核心数据，如员工信息、部门信息、工资记录等，确保数据的准确性和完整性。同时，管理员也需要定期进行数据备份和恢复操作，以保障数据安全。

系统监控和维护：管理员需要监控系统的运行状态，及时处理系统异常情况和故障，确保系统的稳定运行。此外，管理员还需定期对系统进行性能优化和升级，以提升用户体验和系统可靠性。

普通用户：

查看员工信息：普通用户可以通过系统查看员工的基本信息，包括姓名、性别、职位、联系方式等，以便了解员工情况和进行沟通联系。

提交申请：普通用户可以通过系统提交加班、请假、调岗等申请，填写申请表单并提交审批。

用户可以选择申请类型、填写申请原因和期间，并上传相关附件。

查看工资条：普通用户可以查看个人的工资条和工资明细，包括工资基本信息、奖金、扣款等内容，以了解个人的薪资情况和工资结构。

人事部门：

人员管理：人事部门负责管理员工的入职、转正、离职等流程，包括审核招聘需求、发布招聘信息、面试录用、办理入职手续等操作。

薪资管理：人事部门负责管理员工的薪资制度和福利待遇，包括工资标准制定、薪资调整审批、发放工资条等工作。

统计报表：人事部门可以通过系统生成员工信息统计报表、薪资分布报表、人员流动情况报表等，用于人力资源管理和决策分析。

财务部门：

薪资发放：财务部门负责审核和确认员工的工资条，并负责发放工资和奖金。财务部门需要核对工资发放记录，确保工资的准确发放。

成本控制：财务部门需要对员工的薪资支出和福利费用进行统计分析，帮助企业进行成本控制和预算管理。

IT 管理员：

系统安全管理：IT 管理员负责管理系统的安全策略和权限管理，包括用户身份验证、访问控制、数据加密等操作，以保护系统的安全性和稳定性。

系统维护和优化：IT 管理员负责系统的日常维护和优化工作，包括系统安装、配置、升级、性能监控、故障排除等操作，以确保系统的正常运行和高效运行。

3.2.4 关键点

关键功能：

1. 员工信息管理：提供员工基本信息的录入、修改和查询功能，包括姓名、性别、出生日期、联系方式等。
 2. 组织架构管理：支持公司组织结构的管理，包括部门信息、岗位信息等的录入和调整。
 3. 薪资福利管理：实现对员工薪资、奖金、福利等信息的管理和发放，包括工资条生成和查询功能。
 4. 考勤管理：记录员工的考勤情况，包括请假、加班、迟到等，以便进行考勤统计和工资计算。
 5. 绩效评估：实现对员工绩效的评定和考核，包括制定绩效考核计划、评分、排名等功能。
 6. 员工培训管理：记录员工参加的培训活动，包括培训计划、培训内容、培训成绩等。
- 招聘管理：支持招聘流程的管理，包括发布招聘信息、筛选简历、安排面试等。

关键算法：

1. 数据加密算法：用于对敏感数据的加密存储，保障员工隐私和信息安全。
2. 绩效评估算法：设计有效的绩效评价标准和评分标准，确保评估结果客观公正。
3. 薪资计算算法：根据员工的工作时间、考勤情况和绩效表现等因素，自动生成工资条并计算实际工资。
4. 招聘筛选算法：通过智能筛选和匹配算法，提高招聘效率和匹配度，减少人力资源成本。

关键技术：

1. 数据库技术：选择适当的数据库系统，如 MySQL、Oracle 等，存储和管理系统的数据。
2. 前端技术：采用现代化的前端框架和技术，如 React、Angular 等，实现用户友好的界面和交互体验。
3. 后端技术：使用稳定可靠的后端框架和技术，如 Spring Boot、Django 等，实现系统的业务逻辑和数据处理功能。
4. 安全技术：采用 SSL 加密协议、权限控制、防火墙等安全技术，保障系统的数据安全和网络安全。
5. 人工智能技术：应用机器学习和自然语言处理技术，对员工数据进行分析 and 挖掘，提供数据驱动的决策支持。

用户特点：

1. 人力资源部门：负责员工信息的录入、管理和绩效评估等工作，需要系统提供便捷的操作界面和全面的数据统计分析功能。

2. 员工：需要通过系统查询个人信息、查看工资条、提交请假申请等，需要系统界面简洁、操作便捷。

3. 管理层：需要系统提供实时的人力资源数据报表和分析结果，以便进行决策和规划，需要系统界面直观、数据准确。

3.2.5 约束条件

列出进行本系统开发工作的约束条件。例如：经费限制、开发期限和所采用的方法与技术，以及政治、社会、文化、法律等。

经费限制：

开发团队需要在有限的经费内完成软件开发和维护工作。这意味着开发团队需要制定详细的开发计划，合理分配经费，控制开发成本，确保在有限的经费内高效地完成软件开发和维护工作。同时，开发团队还需要考虑经费的投入对软件开发和维护质量的影响，以确保软件的质量和用户满意度。

开发期限：

为了满足市场需求，开发团队需要在约定的时间内完成软件开发，并按时发布上线。开发团队需要根据开发计划合理安排工作进度，提高开发效率，确保软件开发进度符合预期，按时发布上线。同时，开发团队还需要考虑开发期限对软件功能和性能的影响，以确保软件的质量和用户满意度。

技术限制：

开发团队需要使用现有的技术和工具进行软件开发，同时需要遵循相关的技术标准和规范。开发团队需要掌握相关技术和工具的使用方法，熟练掌握开发流程，遵循开发规范和标准，确保软件开发和维护的质量和稳定性。同时，开发团队需要关注技术的更新和发展，及时更新软件的技术和功能，提高软件的竞争力和用户体验。

法律限制：

开发团队需要遵守相关法律法规，如个人信息保护法、著作权法等，确保软件开发和使用符合法律规定。开发团队需要了解相关法律法规，确保软件开发和使用符合法律规定，避免侵犯用户权益和引发法律纠纷。

用户需求限制：

开发团队需要根据用户的需求进行软件开发，确保软件功能和性能满足用户的需求和期望。开发团队需要充分了解用户需求，从用户的角度出发设计软件功能和界面，提高软件的用户体验和满意度。

平台限制：

开发团队需要根据不同的移动设备平台进行开发和适配，确保软件在不同平台上运行稳定和兼容性好。开发团队需要了解不同平台的技术和要求，设计软件的架构和界面，确保软件在不同平台上运行稳定和兼容性好。提高用户使用体验。

数据安全限制：

开发团队需要确保用户上传的音视频和文档数据的安全性，保护用户隐私，防止数据泄露和滥用。对于用户上传的数据，需要采取合理的安全措施进行存储、传输和处理。例如，采用加密技术对数据进行加密存储，采用防火墙和安全认证等措施保护数据传输的安全，对于敏感信息需要采取更严格的访问控制策略，保证用户隐私不被泄露。

可用性限制：

开发团队需要确保软件界面友好、易用，操作简便，遵循用户习惯，提高软件的可用性和用户体验。在开发过程中需要遵循相关的用户界面设计规范和标准，采用合理的图形化和交互式设计，提供适当的帮助和反馈机制，以提高用户对软件的满意度和使用率。同时，需要考虑用户对不同设备和平台的差异，进行适配和测试，确保软件在不同设备和平台上的运行稳定和兼容性好。

敏捷开发：

采用敏捷开发方法，将整个开发过程划分为多个迭代，并在每个迭代中集中精力开发最重要的功能。这种方法可以使开发团队更加专注于实现软件的核心功能，而不是陷入琐碎的细节和不必要的功能中。

开源软件：

考虑使用一些开源软件或开源库来减少开发成本，这些软件和库已经被其他开发者广泛使用和测试过，可以节省开发时间和成本。同时，开发者也可以贡献代码给开源社区，从而在社区的支持下快速开发出更加稳定和完善的软件。

互联网审查：

某些国家对互联网上的内容实行审查制度，开发者需要了解当地互联网审查的要求，并确保其软件符合相关的法律和条例。

3.3 需求规格

3.3.1 软件系统总体功能/对象结构

3.3.2 软件子系统功能/对象结构

对每个主要子系统的基本功能模块/对象进行描述，包括结构图、流程图或对象图。

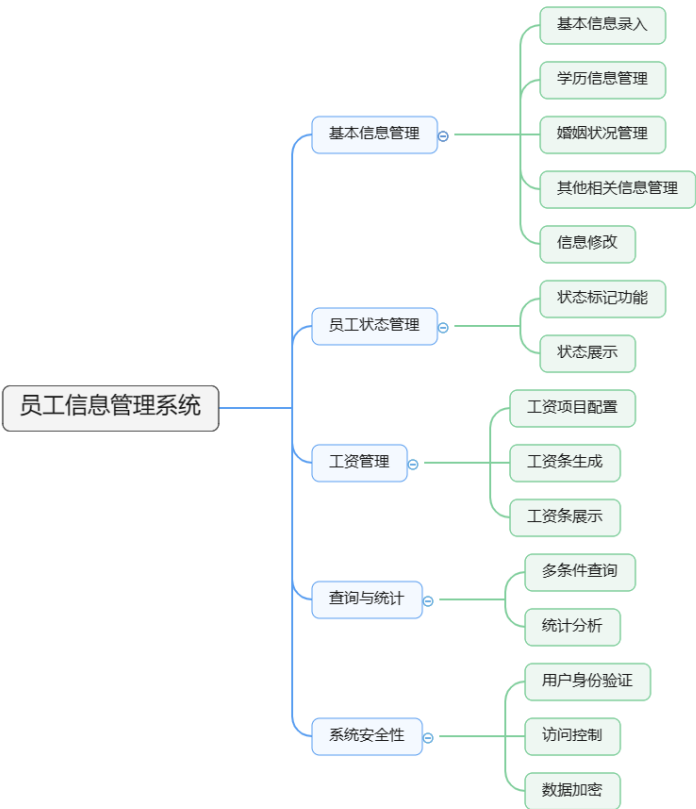


图 1 基本功能模块结构图

3.3.3 描述约定

通常使用的约定描述(数学符号、度量单位等)。

信息格式转换模块

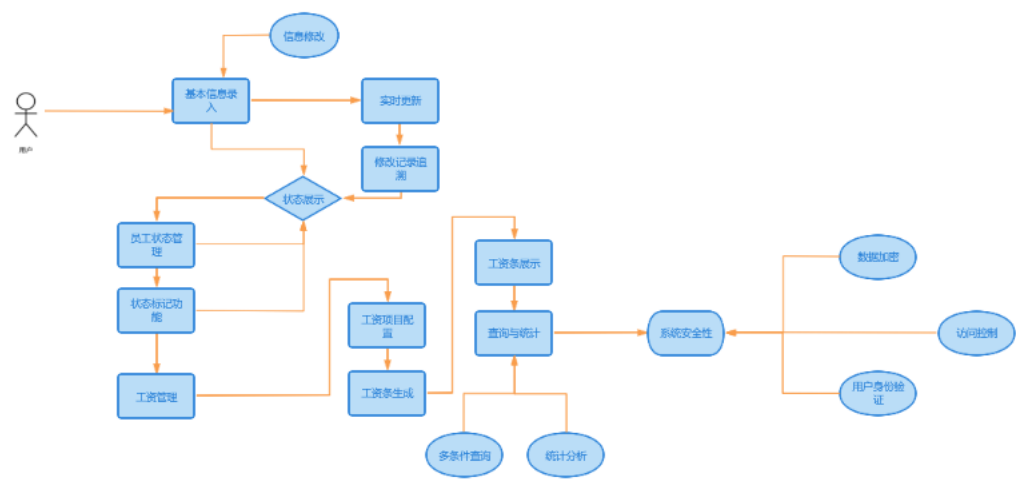


图 2 信息格式转换图

在本需求规格说明中，我们遵循以下描述约定：

- 所有的度量单位均为国际标准单位（SI 单位）。
- 所有的缩写词和首字母缩略词在第一次出现时均需解释其含义。
- 本文档中使用的所有图形符号和缩略语的含义将在每个图形下方进行说明。
- 本文档中使用的所有术语遵循业界通用的定义，并在术语表中进行解释。

3. 4CSCI 能力需求

针对该项目，我们可以将它划分为以下几个 CSCI：

前端应用程序：包括用户界面、查询模块、用户管理模块和社区模块等。

后端应用程序：包括查询模块、和用户管理模块等。

数据库管理系统：负责管理用户信息、数据的存储和检索。

对于每个 CSCI，我们可以定义相应的能力需求，以确保其实现符合预期。以下是该项目的 CSCI 能力需求：

1. 前端应用程序：

用户界面应该清晰、易于使用，支持响应式设计，适配不同的设备和屏幕尺寸。

搜索模块应该能够快速响应用户提交申请及用户输入的搜索请求（员工信息，工资条，违规记录等），并返回相应的搜索结果。

用户管理模块应该提供用户注册、登录和管理员注册、登录、管理功能，支持安全的身份验证和密码加密。

前端应用程序应该与后端应用程序和数据库管理系统进行交互，通过 API 接口传递数据和消息。

2. 后端应用程序：

搜索模块应该能够从该系统数据库中检索用户、管理员要查询的相关信息

用户管理模块应该能够管理用户信息（工资，违规，绩效）和用户状态（升降职，聘用、解聘等）等数据，支持数据加密和安全的存储和检索。

后端应用程序应该与前端应用程序和数据库管理系统进行交互，通过 API 接口传递数据和消息。

3. 数据库管理系统：

数据库管理系统应该支持可扩展性和高可用性，能够处理大量用户和数据。

数据库管理系统应该能够存储和检索用户信息等数据，并支持事务和数据一致性。

数据库管理系统应该能够与前端和后端应用程序进行交互，通过 API 接口传递数据和消息。

3.5 CSCI 外部接口需求

1. 用户登录接口：提供用户登录账号渠道，为保证软件的高可用度，应至少支持手机登录、邮箱登录、游客登录四种形式
2. 用户注册接口：提供用户注册渠道。用户需通过手机号码登录。
3. 员工信息查询：普通用户可以查询员工基本信息，包括姓名、性别、职位、联系方式等，以便了解员工情况和进行沟通联系。普通用户可以查看个人的工资条和工资明细，包括工资基本信息、奖金、扣款。
4. 管理员工接口：管理员拥有对系统各项功能进行配置和管理的权限，包括添加、编辑、删除用户账号，设置用户权限，配置系统参数等。 管理员负责管理系统中的核心数据，如员工信息、部门信息、工资记录等，确保数据的准确性和完整性。
5. 提交申请接口：普通用户可以通过系统提交加班、请假、调岗等申请，填写申请表单并提交审批。用户可以选择申请类型、填写申请原因和期间，并上传相关附件。

3.5.1 接口标识和接口图

列举系统主要功能接口如下：

1. 用户登录接口

(1) 手机登录：Log_Phone(phone, password[, countrycode, md5_password, captcha])

用户需至少提供手机号码、密码，另外可选择传递国家码、MD5 加密后的密码和验证码

接口示例：/login/cellphone?phone=xxx&password=yyy

(2) 邮箱登录：Log_Email(email, password[, md5_password])

用户需至少提供 163 邮箱号、密码，可选择传入 MD5 加密后的密码

接口示例：/login?email=xxx@163.com&password=yyy

接口示例：/login/qr/create?key=xxx

2. 用户注册接口：Register(captcha, phone, password, nickname[, coutrycode])

用户至少要提供验证码、手机号码、密码和昵称，可选择提供国家码(默认是中国)

接口示例：

/register/cellphone?phone=13xxx&password=xxxxx&captcha=1234&nickname=binary1345

3. 员工信息查询接口: Search(keyword[, limit, offset, type])
用户至少需要提供检索关键词（姓名，年龄，入职时间等），同时也可以提供要求返回的数量，偏移量，搜索类型。
接口示例: /search?keywords=张三
4. 管理员工接口: Manage_employee(id)
用户至少需要提供检索关键词（姓名，年龄，入职时间等），对该员工的信息进行修改。
接口示例: /search?keywords=张三
5. 提交申请接口: Submit_applications()
用户通过系统提交加班、请假、调岗等申请，填写申请表单并提交审批。用户可以选择申请类型、填写申请原因和期间，并上传相关附件。

3. 6CSCI 内部接口需求

表 2 CSCI 内部需求表

功能模块	函数名称	接口用途
员工信息管理	addEmployee	添加新员工到系统
	deleteEmployee	从系统中删除指定员工
	changeEmployee	更新员工信息，包括姓名、部门、职位等
	getEmployee	根据 Employee_id 查询员工信息
部门管理	addDepartment	添加新部门到系统中
	deleteDepartment	从系统中删除指定的部门
	updateDepartment	更新部门信息
	getDepartment	根据 Department_id 查询部门
职位管理	addPosition	添加新职位到系统中
	deletePosition	从系统中删除职位
	updatePosition	更新职位信息
	getPosition	根据 Position_id 查询职位信息
员工表现管理	addPerformance	添加员工绩效评定记录
	getPerformance	获得员工的评定记录
工资名单管理	addPayroll	添加一个工资条信息
	getPayroll	获得相应工资条信息
功能模块	url 路径	接口用途
用户（管理员）模块	register_confirm/ (register2_confirm/)	根据 id 查找用户，检查用户是否已经存在
	register/ (register2/)	添加用户，传递必要参数

	login/(login2)	登录接口
--	----------------	------

3.7CSCI 内部数据需求

本条应指明对 CSCI 内部数据的需求，(若有)包括对 CSCI 中数据库和数据文件的需求。如果所有有关内部数据的决策都留待设计时决定，则需在此说明这一事实。如果要强加这种需求，则可考虑在本文档的 3.5.x.c 和 3.5.x.d 给出的一个主题列表。

3.7.1 静态数据

表 3 静态数据表

名称	定义	类型
员工信息表	包括员工的基本信息、教育背景、工作经历、技能证书等。	数据表
薪酬福利表	包括工资结构、福利政策、奖金分配等	数据表
考勤记录表	记录员工的出勤情况、请假记录、加班时长等。	数据表
绩效评估表	记录员工的工作表现、目标完成情况、绩效反馈等。	数据表
劳动合同表	记录晋升、调动、离职等人事变动信息。	数据表
人事变动表	记录合同期限、续签条件、解除条款等。	数据表

3.7.2 动态数据

表 4 动态数据表

名称	定义	类型
人力资源报告	根据员工信息、薪酬福利、考勤记录、绩效评估等数据生成的各种报告和分析	数报告、分析结果
工资发放记录	记录员工过去一年或更久的工资发放情况。	记录
员工个人信息	输出员工的所有详细个人信息。	记录、数据表

月度考勤记录表	输出员工某个月的考勤记录情况	记录
---------	----------------	----

3.8 适应性需求

3.8.1 安装数据要求

数据库：MySQL 数据库，用于存储系统所需的各种静态和动态数据。

Python 环境：Python 3.x 版本，用于运行 Django 框架。

Django 框架：Django 框架提供了强大的 Web 开发功能，用于快速开发和部署人事管理系统。

3.8.2 运行参数要求

数据库连接参数：MySQL 数据库的连接地址、用户名、密码等。

Web 服务器配置：Django 框架的运行端口、域名、静态文件路径等。

系统参数配置：系统日志级别、调试模式开关、国际化语言设置等。

人事管理系统（网站）的运行参数要求包括以下几个方面：

硬件要求：服务器的 CPU、内存、硬盘空间等硬件配置需要满足系统运行的基本要求，以保证系统的流畅运行和数据处理能力。

软件兼容性：系统需要与操作系统兼容，同时可能需要与其他企业管理软件如财务系统、考勤系统等进行集成，因此需要考虑软件之间的兼容性。

网络环境：一个稳定且安全的网络环境是必要的，以确保数据可以实时同步和备份，同时保障数据传输的安全性。

数据安全：系统应具备数据加密、备份和恢复等功能，以防止数据丢失或被非法访问。

用户界面：系统的操作界面应友好易用，以便员工能够快速上手，提高工作效率。

综上所述，人事管理系统的运行参数要求涉及多个方面，企业在选择时应根据自身的具体需求和预算进行综合考虑。此外，随着人力资源数字化转型的不断深入，一体化的 HR SaaS 解决方案越来越受到企业的青睐，因为它们能够提供更加全面和智能化的人才管理数字基础设施。在考虑人事管理系统时，企业也应关注系统的数字化和智能化程度，以及是否能够支持组织的长期发展和变革需求。

3.9 保密性需求

3.9.1 用户隐私

未来保护用户隐私安全，传输用户数据时使用了 AES 加密算法，且密码等保密级别极高的数据不在数据库中明文存储。

3.9.2 验证系统

用户必须提供正确的用户名和密码才能登录系统。同时，系统应该提供多种身份验证机制，如短信验证码等，以确保用户的身份得到有效验证。同时由于不同身份的用户对于不同的接口使用权限不同，应有完善的验证系统对不同用户的身份进行检查。保证普通用户，管理员用户可以分别执行对应的操作，确保系统的有序性。

3.9.3 防止 SQL 注入

SQL 注入会对后台数据库造成很大的威胁，也增大了用户信息泄露的危险。因此拥有防止 SQL 注入的机制是必要的。该项目通过采用 ORM 等机制来解决 SQL 注入的威胁，保证数据库的安全性。

3.9.4 XSS 保护

本项目中有很多需要提交的表单，如果这些地方被植入脚本，很可能对系统造成威胁，所以需要 XSS 防范机制。

需要使用 XSS 防护机制来对用户可操纵的数据进行 HTML 转义，之后再呈现给用户。如果用户可操纵的内容是 HTML 的标签属性，那么还要进行关键词的黑名单/白名单的过滤。

3.9.5 完善日志系统

项目建立实时记录应用行为的运行日志，建立完善的日志系统，当应用出现问题时能够帮助我们发现、解决问题。

3.9.6 数据库可恢复

需要定期为数据库进行备份，将数据库文件数据备份到另一台服务器。

3.10 保密性和私密性需求

人事管理系统非常注重用户数据的保密性和私密性，并采取了多项措施来确保用户数据的安全。其中，账号密码和短信验证相结合的身份验证方式可以有效防止未授权的访问，而对数据进行 AES 加密可以保证传输过程中的数据不被窃取。同时，定期备份的方式也可以在数据丢失或损坏时恢复数据。此外，采用 ORM 机制、XSS 防范机制及黑白名单的方式可以有效预防 SQL 注入和脚本注入等攻击方式。人事管理系统还遵循了《网络安全法》、《个人信息保护法》等隐私政策和数据安全要求，并符合 ISO 27001 安全认证标准，以确保安全性和隐私性符合标准和规范。

3.11CSCI 环境需求

表 5 环境需求表

类别	环境	计算机硬件需求	操作系统
服务器	交互服务器	CPU：4 核及以上； 内存：8G 及以上； 硬盘：1TB 及以上	Windows Server 2012 及以上、 Linux
	GPU 服务器	CPU：8 核及以上； 内存：32G 及以上； 硬盘：2TB 及以上； GPU：Nvidia Tesla V100/ A100/ P100	Windows Server 2012 及以上、 Linux
	文件服务器	CPU：4 核及以上； 内存：8G 及以上； 硬盘：2TB 及以上	Windows Server 2012 及以上、 Linux
客户端	Web 客户端	CPU：双核及以上； 内存：1G 及以上； 硬盘：64GB 及以上	Andriod、Linux、 Windows7 及其以上

3.12 计算机资源需求

3.12.1 计算机硬件需求

3.12.2 计算机硬件资源利用需求

表 6 计算机硬件资源利用需求表

类型	最大许可使用的处理器能力	存储器容量	输入/输出设备能力	辅助存储器容量	通信/网络设备能力	资源利用的条件
交互服务器	至少 1 颗处理器，速度不低于 2GHz	至少 8GB	至少 1 个显示器和 1 个键盘	至少 500GB	至少 1 个网络接口卡	每个硬件资源的利用率不超过 80%
GPU 服务器	至少 1 颗处理器，速度不低于 2GHz	至少 8GB	不需要	至少 500GB	至少 1 个网络接口卡	每个硬件资源的利用率不超过 80%
文件服务器	至少 1 颗处理器，速度不低于 2GHz	至少 8GB	不需要	至少 500GB	至少 1 个网络接口卡	每个硬件资源的利用率不超过 80%
Web 客户端	至少 1 颗处理器，速度不低于 2GHz	至少 8GB	不需要	不需要	至少 1 个网络接口卡	每个硬件资源的利用率不超过 80%

3.12.3 计算机软件需求

表 7 计算机软件需求表

环境	操作系统	数据库管理系统	通信/网络软件	实用软件	输入和设备模拟器	测试软件	生产用软件

服务器	Ubuntu 20.04 LTS	MySQL 8.0.23	Apache HTTP Server 2.4.46	OpenSSH 8.2p1	无	LoadRunner	无
Web 客户端	windows	无	HTTP	Vscode	Web Emulator	SVN	无

3.12.4 计算机通信需求

表 8 计算机通信需求

连接地理位置	配置和网络拓扑结构	传输技术	数据传输速率	网关	系统使用时间要求	传送/接收数据的类型和容量	传送/接收/响应的时间限制	数据的峰值	诊断功能
外部 Web 服务器、数据库服务器、云存储服务	局域网，使用 TCP/IP 协议，双向连接，CSCI 为客户端	HTTP、HTTPS 协议	100Mbps	不需要	周一至周五，上午 9 点至下午 5 点	JSON 格式数据，图片文件（大小不超过 10MB）	100ms	每小时 1000 次数据传输的峰值	提供诊断功能，进行故障排除

3.13 软件质量因素

本节描述了本系统的软件质量要求，包括有关 CSCI 的功能性、可靠性、可维护性、可用性、灵活性、可移植性、可重用性、可测试性、易用性以及其它属性的定量需求。本节描述了本系统的软件质量要求，包括有关 CSCI 的功能性、可靠性、安全性、可用性、灵活性、可移植性、可重用性、可测试性、适应性以及其它属性的定量需求。

3.13.1 功能性：

软件质量因素对于评估软件系统至关重要。针对我们的人事管理系统，功能性是其中一个至关重要的评估指标，因为它确保系统能够有效地实现所有在合同中规定的功能需求，以支持高效的人力资源管理。

首先，系统必须能够完整地实现所有在合同中规定的功能模块。这包括员工信息管理、信息修改、员工状态管理、工资管理、查询与统计，员工提交申请以及系统安全性等功能。系统应具有一定的可扩展性和可配置性，以便在今后的开发中能够轻松地添加新的功能模块，以满足不断变化的人力资源管理需求。

其次，系统需要准确地解释和处理输入数据。这意味着系统必须能够正确地识别和解析各种类型的员工信息、工资数据等，确保在后续处理中能够准确地匹配和处理这些数据。

最后，系统必须能够产生准确、一致的输出结果。在查询与统计功能中，系统需要能够根据不同条件查询员工信息和工资数据，并生成准确的统计报表。在系统安全性方面，系统需要具备用户认证和权限管理功能，以确保只有授权用户能够访问系统，并保护员工信息的安全。

综上所述，功能性对于我们的人事管理系统至关重要。系统必须能够完整地实现所有功能模块，并且能够正确解释和处理输入数据，以产生准确、一致的输出结果，从而支持有效的人力资源管理。

因此，本系统的功能性要求如下：

- 系统满足合同中规定的所有功能需求。
- 系统正确地解释和处理输入数据，并产生正确、一致的输出结果。
- 系统提供易于使用的用户界面，以使用户能够轻松地使用系统的所有功能。
- 系统支持所有必要的数据输入和输出格式，并提供必要的的数据转换和兼容性支持。
- 系统提供适当的错误处理和恢复机制，以确保系统的可靠性和稳定性。
- 系统提供适当的性能和响应时间，以确保满足用户需求。
- 系统支持并遵循相关的行业标准和规范，以确保系统的可靠性和互操作性。

3.13.2 可靠性：

本系统具有高可靠性。系统能够在正常操作下正常运行，而不会崩溃或出现意外错误。系统在处理异常情况时能够正确地处理，并能够尽快地恢复正常状态。

为了确保我们这款 app 具有高可靠性，我们需要在开发和测试过程中采取一系列措施来预防和处理异常情况。这些措施包括但不限于：

- 异常处理：我们需要在代码中加入适当的异常处理机制，以确保在出现异常情况时，系统能够正确地处理并进行相应的提示和恢复操作。
- 数据备份：在用户上传音视频或文档时，我们需要及时进行数据备份，以保证在系统出现故障时，能够快速恢复数据，避免数据丢失。
- 测试覆盖率：我们需要对系统进行全面的单元测试和集成测试，并且要确保测试覆盖率达到预期，以尽量减少系统的潜在缺陷和错误。
- 可靠性测试：我们需要对系统进行可靠性测试，以模拟系统在不同负载和场景下的运行情况，确保系统能够在各种情况下正常运行并保持稳定。
- 日志记录：我们需要对系统进行日志记录，以便在出现问题时能够快速

地定位和解决问题，并对系统进行优化改进。

通过以上措施的实施，我们可以提高系统的可靠性，确保系统能够在正常操作下正常运行，并在出现异常情况时能够快速地恢复正常状态。

3.13.3 安全性：

在我们的人事管理系统中，安全性也是其中一个至关重要的评估指标。它确保系统能够有效地保护员工信息和系统数据的安全性，从而维护人力资源管理的可信度和隐私保护。

首先，系统必须确保具备可靠的认证和权限管理功能。这意味着只有经过授权的用户才能够访问系统，而系统应根据用户角色和权限设置不同的访问级别。

其次，系统应具备完善的日志记录和审计功能。通过记录用户的登录和操作日志，以及监测系统的安全事件和异常活动，系统能够及时发现潜在的安全风险和问题，并采取相应的措施进行处理。

最后，系统必须采用可靠的数据加密技术来保护敏感信息的安全。对于员工个人信息、工资数据等敏感数据，系统应采用加密存储和传输的方式，以防止数据泄露和未经授权的访问。

综上所述，安全性是我们人事管理系统中不可或缺的一环。系统的认证和权限管理、日志记录和审计功能以及数据加密技术都是确保人力资源管理的可信度和隐私保护的重要保障。

3.13.4 可用性：

我们的人事管理系统具备了高水准的可用性，为用户提供了便捷、稳定的服务。以下是系统在可用性方面的主要特点：

首先，系统拥有直观、易用的用户界面，使用户能够轻松地完成各项操作，无需繁琐的培训。用户可以快速找到所需功能，并且完成操作所需的步骤简单明了。

其次，系统保持了稳定可靠的运行状态。我们经过了严格的测试和验证，确保系统在各种条件下均能保持高可用性和稳定性，避免因系统故障而导致的服务中断或数据丢失。

另外，我们提供了及时的技术支持和维护服务。用户在使用过程中遇到问题时，可以随时联系我们的技术支持团队，获得专业的帮助和解决方案，以确保系统能够持续稳定地运行。

最后，我们的系统具备灵活性和可扩展性。随着人力资源管理需求的变化，我们可以方便地添加新的功能模块或进行系统升级，以满足不断变化的用户需求，保持系统的竞争力和可持续发展。

综上所述，我们的人事管理系统已经取得了令人满意的成果，在可用性方面表现出色，为用户提供了高质量的服务和良好的用户体验。

3.13.5 灵活性:

我们的人事管理系统展现出了很高的灵活性，能够迎合不断变化的人力资源管理需求。以下是系统灵活性的几个方面：

系统设计灵活性。我们采用了模块化的架构设计，使得系统能够轻松地添加新的功能模块或进行定制化的功能扩展。这样的设计让我们能够根据不同组织的需求定制个性化的解决方案。

参数配置灵活性。系统提供了丰富的参数设置和配置选项，管理员可以根据具体的业务需求和管理策略进行灵活的配置。这使得系统能够根据不同的使用场景进行定制化设置，满足用户的特定需求。

智能化算法和策略。系统利用智能算法对用户行为和数据进行分析，能够不断学习和优化，以更好地匹配用户需求并提供个性化的服务。这使得系统能够随着时间的推移不断适应用户的变化需求，提供更加智能的服务。

集成能力。系统具有良好的集成能力，提供了丰富的 API 接口和标准协议，使得系统能够与其他系统无缝集成。这样一来，系统能够实现与外部系统的数据共享和业务流程的无缝对接，进一步提升了系统的灵活性和整体效能。

综上所述，我们的人事管理系统在设计、配置、算法和集成等方面都表现出了很高的灵活性，能够根据不同的需求进行定制化配置和功能扩展，以应对不断变化的人力资源管理挑战。

3.13.6 可移植性:

我们的人事管理系统展现出了出色的可移植性，能够在不同的平台和环境下灵活运行。以下是系统可移植性的几个方面：

跨平台兼容性。我们的系统采用了跨平台的技术方案，使得系统能够在不同的操作系统上运行，包括 Windows、Linux 和 macOS 等。这意味着用户可以在不同的设备上无缝地访问和使用系统，而无需担心平台兼容性的问题。

数据库独立性。系统与特定的数据库解耦，采用了标准的数据库接口和查询语言，如 SQL，使得系统能够在不同的数据库管理系统上运行，包括 MySQL、PostgreSQL、Oracle 等。这样的设计确保了系统对数据库的依赖性较低，提高了系统的可移植性和灵活性。

软件依赖性管理。我们尽可能地减少了系统对第三方软件和库的依赖，使得系统能够在不同的环境下轻松部署和运行。同时，我们对系统所需的软件和库进行了管理和打包，以便在不同的环境下进行快速部署和安装。

模块化设计。系统采用了模块化的架构设计，将不同的功能模块进行了分离和解耦，使得系统的各个模块可以独立地进行开发、测试和部署。这样的设计提高了系统的可移植性和可维护性，使得系统能够更容易地在不同的环境下进行部署和运行。

综上所述，我们的人事管理系统在跨平台兼容性、数据库独立性、软件依赖性管理和模块化设计等方面都表现出了出色的可移植性。这使得系统能够在不同的平台和环境下灵活运行，为用户提供稳定可靠的服务。

3.13.7 可重用性:

我们的人事管理系统展现出了优秀的可重用性，能够在不同的场景和需求下被灵活地重复利用。以下是系统可重用性的几个方面：

模块化设计。我们的系统采用了模块化的架构设计，将不同的功能模块进行了分离和解耦。这样的设计使得每个模块都可以独立地进行开发、测试和部署，从而提高了模块的可重用性。开发人员可以根据需要单独使用某个模块，或者将模块组合起来构建新的系统，以满足不同的业务需求。

标准化接口。我们的系统提供了标准化的接口和 API，使得不同的系统和应用程序能够与我们的系统进行无缝集成。这样的设计使得系统的功能模块可以被其他系统和应用程序轻松地调用和重用，从而提高了系统的可重用性和整体的可扩展性。

代码复用。我们在系统开发过程中充分利用了现有的代码库和开源组件，避免了重复开发相似功能的情况。这些通用的代码库和组件可以被不同的项目和团队共享和重用，从而提高了开发效率和代码质量。

组件化开发。我们将系统的各个功能模块进行了组件化开发，使得每个组件都可以独立开发、测试和部署。这样的设计使得开发人员可以根据需要选择和组合不同的组件，构建出满足特定需求的系统，从而提高了系统的可重用性和灵活性。

综上所述，我们的人事管理系统在模块化设计、标准化接口、代码复用和组件化开发等方面都表现出了优秀的可重用性。这使得系统的各个功能模块可以被灵活地重复利用，为不同的项目和需求提供了高效可靠的解决方案。

3.13.8 可测试性

我们的人事管理系统在可测试性方面有着显著的优势，这为确保系统的功能完整性和稳定性提供了坚实的基础。以下是系统可测试性的几个亮点：

清晰的需求规格：我们在系统设计阶段就明确定义了系统的功能和性能需求，并将其详细记录下来。这些规格成为测试团队的参考，确保他们能够准确地测试系统的每个方面，包括各项功能和性能指标。

模块化测试：系统采用了模块化的设计架构，每个功能模块都被设计为独立的单元。这种设计使得我们能够针对每个模块进行单独的测试，从而更容易地发现和解决潜在的问题和缺陷。

自动化测试：我们引入了自动化测试工具和框架，对系统的各个功能模块进行了自动化测试。通过自动化测试，我们能够快速地执行大量的测试用例，并及时发现系统中可能存在的问题，提高了测试的效率和覆盖率。

异常情况测试：除了常规功能测试外，我们还对系统进行了异常情况测试，验证系统在面对异常情况时的表现。这种测试能够帮助我们发现系统的潜在薄弱点，并及时进行修复和优化。

持续集成和持续部署：我们采用了持续集成和持续部署的开发模式，将测试环节纳入到持续集成流程中。每次代码提交后，系统都会自动触发测试流程，包括单元测试、集成测试和端到端测试，以确保代码的质量和稳定性。

3.13.9 适应性:

我们的人事管理系统在适应性方面表现出色，能够灵活应对不同用户群体和工作环境的需求。以下是系统适应性的几个方面：

用户定制化：系统提供了灵活参数设置和配置选项，使得管理员可以根据不同用户群体的需求进行个性化定制。无论是大型企业还是中小型公司，都能够根据自身需求进行灵活配置，满足不同规模和业务需求下的人力资源管理需求。

多平台兼容性：我们的系统具备多平台兼容性，可以在各种不同的操作系统和设备上运行，包括 Windows、Linux、iOS 和 Android 等。这使得用户可以在不同的工作环境和设备上方便地使用系统，提高了系统的适应性和灵活性。

多语言支持：为了满足全球化的需求，我们的系统提供了多语言支持，用户可以根据自己的偏好选择系统界面的语言。这种多语言支持使得系统能够适应不同国家和地区的用户群体，提高了系统的用户友好性和可用性。

弹性扩展性：我们的系统具备弹性扩展性，可以根据业务需求和用户规模进行灵活扩展。无论是人员增加还是业务扩展，系统都能够随着需求的变化进行相应的扩展和优化，确保系统始终能够满足用户的需求。

自适应界面：系统设计了响应式界面，能够根据用户使用的设备和屏幕尺寸进行自适应调整，确保在不同设备上都能够提供良好的用户体验。这种自适应界面使得用户无论在台式电脑、笔记本电脑还是移动设备上都能够方便地访问和使用系统。

3.14 设计和实现的约束

人事管理系统需要采用现代化的编程语言和开发框架进行开发，以确保软件的稳定性和可扩展性。

人事管理系统需要实现高效的员工信息管理、信息修改、员工状态管理、工资管理、查询与统计以及系统安全性等功能，以支持有效的人力资源管理。

为了确保系统对各种输入数据的准确性和高质量的处理，人事管理系统需要使用先进的算法和技术，如数据挖掘、机器学习和自然语言处理等，以实现员工信息的准确提取和处理。

人事管理系统需要采用云存储服务，以确保数据的高可靠性和备份恢复能力。这样可以保证员工信息和相关数据的安全性和可靠性。

为了保证软件开发的高质量和及时性，人事管理系统需要采用合适的开发模式和流程，如敏捷开发、代码审查、测试等。这样可以确保系统的稳定性和功能完整性，并及时满足用户需求的变化。

3.15 数据

说明本系统的输入、输出数据及数据管理能力方面的要求(处理量、数据量)。

3.15.1 输入数据

1. **员工信息**：包括员工的基本信息、教育背景、工作经历、技能证书等。
2. **薪酬福利**：工资结构、福利政策、奖金分配等。
3. **考勤数据**：员工的出勤情况、请假记录、加班时长等。
4. **绩效评估**：员工的工作表现、目标完成情况、绩效反馈等。
5. **人事变动**：包括晋升、调动、离职等人事变动信息。
6. **劳动合同**：合同期限、续签条件、解除条款等。

3.15.2 输出数据

1. **报告和分析**：生成各种人力资源报告，如人员结构报告、薪酬分析报告、绩效统计报告等。
2. **工资记录**：输出员工过去一年或更久的工资发放记录情况。
3. **员工信息**：输出员工的所有详细个人信息。
4. **考勤记录**：输出员工某个月考勤记录情况

3.15.3 数据管理

- **存储**：系统提供数据存储功能，将用户上传的信息存储在人事管理系统数据库中。
- **检索**：系统支持数据检索功能，用户可以根据关键字（姓名，年龄，入职时间等）、分类等方式查询相关数据。
- **处理**：系统提供数据处理模块，将用户输入的信息录入系统，对用户要求的信息从数据库中取出进行处理，生成各种报表等输出数据。
- **规模**：根据公司规模，人事管理功能复杂程度而定。
- **备份和恢复机制**：系统有定期备份机制和数据恢复功能，确保数据安全和可靠性。
- **数据安全性要求**：数据库中的信息采取加密措施保护数据安全。

3.15.4 数据处理

- **类型和规模**：系统需要处理的数据类型包括员工或管理员提交的信息表单，规模根据表单数据量而定。
- **速度、准确性和可靠性要求**：处理速度快速响应，准确性和可靠性保证

- **算法和规则：**系统采用先进的生成图表等技术算法和规则，确保数据处理的正确性和可靠性。

3.16 操作

说明本系统在常规操作、特殊操作以及初始化操作、恢复操作等方面的要求。

3.16.1 常规操作

用户可以在浏览器访问人事管理系统网站，注册登录后通过操作界面实现员工基本信息查询，个人信息查询修改，提交申请，查看工资条，考勤，绩效具体内容；管理员用户注册登录后通过操作界面实现员工信息查询修改，管理员工状态，管理考勤绩效内容，生成报告等。

3.16.2 特殊操作

当网站遇到故障、数据损坏或网络断开等情况时，采取以下措施：

立即备份：如果有备份机制，确保立即进行数据备份，以防数据进一步损坏或丢失。

诊断问题：尝试诊断问题的原因，是系统软件故障、硬件问题还是网络服务供应商的问题。

联系技术支持：如果自行诊断无法解决问题，应立即联系系统的技术支持服务，寻求专业帮助。

通知用户：及时通知系统用户，告知他们系统存在的问题和预计的解决时间，以减少对工作的影响。

切换到备用系统：如果有备用的人事管理系统，应立即切换过去，以保证工作的连续性。

记录事件：详细记录发生的故障和处理过程，以便事后分析和预防类似事件再次发生。

检查数据恢复方案：如果数据损坏严重，需要检查并执行数据恢复计划，以尽可能恢复丢失的数据。

加强监控和预防：在问题解决后，加强对系统的监控，并采取措施预防未来的故障，如升级硬件、更新软件、增强网络安全等。

3.16.3 初始化操作

系统设置：首先需要对系统的基本参数进行设置，包括公司信息、部门结构、职位设置等。这些信息将作为系统运行的基础数据。

用户账户创建：为公司员工创建账户，并分配合适的权限。这通常涉及到添加员工的基本信息，如姓名、职位、联系方式等，并设置其对系统的访问权限。

数据导入：如果之前有使用其他人事管理工具，需要将旧系统中的数据迁移到新系统中。这可能包括员工的个人资料、薪酬记录、考勤数据等。

功能模块配置：根据企业的具体需求，配置系统内的各个功能模块，如招聘管理、培训管理、绩效考核等。

流程定义：在系统中定义和配置各项业务流程，包括审批流程、报告生成流程等，以确保系统能够按照企业的运营方式进行运作。

测试验证：在系统初始化完成后，进行全面的测试，确保所有功能都能正常运行，数据准确无误，并且用户能够顺利地各项操作。

培训与上线：对员工进行系统操作培训，确保他们了解如何使用新系统。培训完成后，可以将系统投入实际运营。

3.16.4 恢复操作

如果系统遭遇数据损坏或者其他问题导致无法正常运行，可以使用备份数据进行恢复操作。系统可以按照帮助文档中提供的指南进行恢复操作，也可以联系技术支持团队获取进一步帮助。在进行恢复操作之前，需要确保备份数据的完整性和正确性，以避免数据丢失和恢复失败。

3.17 故障处理

3.17.1 定义故障类型和级别

本系统可能出现的故障类型和级别如下：

- 硬件故障：严重故障
- 软件故障：重要故障
- 网络故障：一般故障

3.17.2 设计故障处理策略

针对不同类型和级别的故障，我们设计相应的故障处理策略，包括以下环节：

- 故障检测：系统能够检测到可能出现的故障类型和级别，例如硬件故障、软件故障、网络故障等，并能够及时发出警报和通知相关人员。
- 故障诊断：一旦出现故障，我们及时进行故障诊断，确定故障原因和影响范围，并及时通知相关人员进行处理。
- 故障处理：针对不同类型和级别的故障，我们采取相应的处理措施，例如重新启动系统、更换硬件设备、修复软件代码等。
- 故障修复：针对一些比较严重的故障，我们及时进行修复，以确保系统能够尽快恢复正常运行状态。

我们明确故障处理的责任和流程，包括故障报告、故障跟踪、故障记录、故障分析等。

3.17.3 提供故障处理支持

我们提供故障处理支持，包括以下方面：

- 故障处理文档：我们提供故障处理文档，详细介绍故障处理策略、流程和各个环节的责任。
- 故障处理工具：我们提供故障处理工具，帮助相关人员进行故障检测、诊断、处理和修复等工作。
- 故障处理培训：我们对相关人员进行故障处理培训，提高其故障处理能力和水平。

我们明确故障处理的时间限制和服务水平，以确保及时、有效地处理故障。在设计故障处理策略时，充分考虑系统的可靠性和可维护性，采取相应的措施提高系统的故障容忍度和故障修复能力，以确保系统在故障情况下的正常运行和恢复。此外，还定期对系统进行故障模拟和测试，以检验故障处理策略的有效性和可靠性。

3.18 算法说明

(1) 搜索方法：

线性搜索算法：

线性搜索算法是一种基本的搜索技术，它顺序地遍历数据集，逐个检查每个元素，直到找到目标元素或者遍历完整个数据集。

$$\text{LinearSearch}(arr, x) = \begin{cases} i & \text{if } arr[i] = x \text{ for some } i \\ -1 & \text{otherwise} \end{cases}$$

B-树算法：

B-树是一种自平衡的搜索树结构，用于组织数据以支持高效的搜索、插入和删除操作。在 B-树中，每个节点都可以包含多个子节点，并且具有固定的最小和最大子节点数。

$$\text{BTreeSearch}(R, x) = \begin{cases} R & \text{if } R \text{ is leaf and contains } x \\ \text{BTreeSearch}(\text{Child}(R, i), x) & \text{if } R \text{ is not leaf and } x \text{ belongs to child } i \\ \text{null} & \text{if } x \text{ is not found in } T \end{cases}$$

哈希表算法：

哈希表是一种基于哈希函数实现的数据结构，用于存储键值对。通过哈希函数，将键映射到表中的特定位置，从而实现快速的搜索、插入和删除操作。

$$\text{HashTableSearch}(T, x) = \begin{cases} \text{value} & \text{if } T[h(x)] = \text{value} \\ \text{null} & \text{if } x \text{ is not found in } T \end{cases}$$

查询与统计：

多条件查询：

在 MySQL 中，可以使用 SELECT 语句来进行多条件查询。多条件查询可以使用 AND、OR、NOT 等逻辑运算符来连接多个条件。下面是一个简单的多条件查询的示例：

假设我们有一个名为 `employees` 的表，包含以下字段：`id`, `name`, `age`, `department`。

1. 使用 `AND` 运算符查询年龄在 25 到 35 岁之间，并且部门是“IT”的员工：

```
SELECT * FROM employees

WHERE age BETWEEN 25 AND 35

AND department = 'IT';
```

2. 使用 `OR` 运算符查询年龄在 25 到 35 岁之间或者部门是“IT”的员工：

```
SELECT * FROM employees

WHERE (age BETWEEN 25 AND 35)

OR department = 'IT';
```

3. 使用 `AND` 和 `OR` 运算符结合查询年龄在 25 到 35 岁之间，并且部门是“IT”或者“HR”的员工：

```
SELECT * FROM employees

WHERE (age BETWEEN 25 AND 35)

AND (department = 'IT' OR department = 'HR');
```

4. 使用 `NOT` 运算符查询年龄不在 25 到 35 岁之间的员工：

```
SELECT * FROM employees

WHERE NOT (age BETWEEN 25 AND 35);
```

可以根据自己的需求组合使用这些条件来完成复杂的查询。

统计分析：

在我们的工程里，Django 框架中进行统计分析并可视化数据通常包括以下步骤：

1. 从数据库中获取数据：使用 Django 的模型（Model）与 ORM（对象关系映射）功能，从数据库中检索数据。我们可以编写自定义的查询来过滤需要的数据。

```
from myapp.models import YourModel

# 从数据库中获取需要的数据

queryset = YourModel.objects.filter(条件)
```

2. 进行统计分析：使用 Python 的统计分析库（例如 Pandas、NumPy 等）对获取的数据进行

统计分析，包括求和、平均值、中位数、标准差等。

```
import pandas as pd

# 将查询结果转换为 DataFrame

df = pd.DataFrame(list(queryset.values()))

# 进行统计分析

result = df.groupby('某个字段').agg({'另一个字段': ['sum',
'mean', 'median', 'std']})
```

3. 处理后的数据放到定义好的表单中，在 `views.py` 中，当用户提交表单时，根据请求的方法（GET 或 POST）来渲染表单页面或处理表单数据。

```
# 在应用的 views.py 文件中定义视图函数

from django.shortcuts import render

from .forms import YourForm

def your_view(request):

    if request.method == 'POST':

        form = YourForm(request.POST)

        if form.is_valid():

            # 处理表单数据

            name = form.cleaned_data['name']

            email = form.cleaned_data['email']

            message = form.cleaned_data['message']

            # 这里可以将数据保存到数据库中，也可以进行其他操作

        else:

            form = YourForm()

    return render(request, 'your_template.html', {'form':
form})
```

html 中使用表单的代码如下：

```
<!-- 在模板文件中显示表单 -->

<form method="post">

    {% csrf_token %}

    {{ form.as_p }}

    <button type="submit">提交</button>

</form>
```

系统安全性:

用户身份验证:

Django 框架使用基于 session 的用户身份验证系统，包括以下几个部分:

用户模型: Django 的 `django.contrib.auth.models.User` 模型用于存储用户的基本信息，如用户名、密码（哈希后的）、电子邮件等。

认证后端: Django 提供了多种认证后端，用于验证用户的身份。默认情况下，Django 使用 `django.contrib.auth.backends.ModelBackend`，该后端通过查询数据库中的用户模型来验证用户的身份。

表单和视图: Django 提供了用于用户登录、注册、注销等操作的表单和视图。你可以使用 `django.contrib.auth.forms` 中的表单类来构建登录和注册表单，使用 `django.contrib.auth.views` 中的视图来处理用户的认证请求。

Sessions: Django 使用 Sessions 来跟踪用户的登录状态。当用户成功登录后，Django 会在服务器端创建一个 Sessions，并将 SessionsID 存储在用户的浏览器的 cookie 中。用户的 Sessions 数据将被存储在服务器端的数据库中，默认情况下是使用数据库后端存储会话数据。

装饰器和 Mixin 类: Django 提供了装饰器和 Mixin 类来保护视图，只允许已登录的用户访问特定的页面。

用户身份验证流程如下:

用户访问需要身份验证的页面，系统检查用户的会话，如果会话存在且有效，则用户被视为已经登录。

如果用户尚未登录，则系统将用户重定向到登录页面。

用户填写登录表单并提交，系统验证用户提供的凭据。

如果凭据验证成功，系统将创建一个新的会话，并将用户重定向到登录前的页面。

如果凭据验证失败，系统将显示错误消息，并保持用户在登录页面。

数据加密:

在 Django 框架中，用户密码存储使用了哈希算法进行加密，而不是使用对称加密或者非对称加密算法。Django 使用了密码哈希算法对用户密码进行加密存储，常用的哈希算法包括 MD5、SHA-1、SHA-256 等。推荐使用 PBKDF2、BCrypt 或者 Argon2 这样的密码哈希算法。默

认使用的是 PBKDF2 算法。

PBKDF2 (Password-Based Key Derivation Function 2) 是一种基于口令的密钥派生函数, 其主要功能是从一个用户提供的密码和一个盐值生成一个安全的密钥。PBKDF2 算法的安全性建立在哈希函数的安全性和迭代次数的选择上。

PBKDF2 的数学公式如下所示:

其中

PBKDF2 算法的安全性取决于哈希函数的安全性以及迭代次数的选择。增加迭代次数可以增加攻击者破解密码的成本。

可以通过以下代码来控制 PBKDF2 算法的迭代次数:

```
# settings.py

PASSWORD_HASHERS = [

    'django.contrib.auth.hashers.PBKDF2PasswordHasher',

    ...

]

# PBKDF2 迭代次数

PBKDF2_ITERATIONS = 150000
```

3.19 有关人员需求

为了保证系统的正常运行和维护, 需要一定数量的人员来支持开发、测试、运维、客服等工作。具体需求如下:

3.19.1 开发团队

开发团队由若干软件开发工程师组成, 负责人事管理系统的设计、开发和维护。开发团队具备以下能力和素质:

- 熟练掌握相关技术和工具, 包括但不限于图表生成、前端和后端开发等。
- 具有良好的编码风格和注释习惯, 能够编写高质量的代码和文档。
- 具备团队协作和沟通能力, 能够与产品经理、测试工程师、运维工程师等紧密合作,

共同推动项目的进展。

- 具备较强的问题解决能力和创新意识，能够快速定位和解决开发过程中遇到的问题。

3.19.2 测试团队

测试团队由若干测试工程师组成，负责对人事管理系统网站进行各类测试，包括但不限于功能测试、性能测试、安全测试等。测试团队具备以下能力和素质：

- 具备较强的测试技能和方法论，能够制定合理的测试计划和测试用例，保证测试的全面和有效。
- 具备严谨的测试思维和分析能力，能够发现和定位各类问题，并对问题进行有效的跟踪和管理。
- 具备与开发团队和产品经理沟通和协作的能力，能够及时反馈测试结果和建议，为产品质量提供保障。

3.19.3 运维团队

运维团队由若干运维工程师组成，负责人事管理系统网站的部署、监控、维护和升级等工作。运维团队具备以下能力和素质：

- 具备较强的服务器管理和网络管理能力，能够保证系统的高可用性和安全性。
- 具备较强的故障诊断和解决能力，能够快速定位和解决各类系统问题。
- 具备与开发团队和测试团队沟通和协作的能力，能够及时响应和处理各类问题和需求。

3.19.4 客服团队

客服团队由若干客服代表组成，负责回答用户的问题和反馈，解决用户的问题并提供良好的用户体验。客服团队具备以下能力和素质：

具备良好的沟通能力和服务意识，能够以友好、耐心、专业的态度解答用户的问题和反馈。熟悉人事管理系统网站的各种功能和使用方法，能够为用户提供准确、及时的帮助和支持。具备较强的问题解决能力和协调能力，能够在最短时间内处理各种用户反馈和投诉。具备积极的学习能力和工作态度，能够不断提升自己的服务水平和用户满意度。

3.20 有关培训需求

为了保证人事管理系统网页的正常运行和维护，需要对相关人员进行培训，以提高其工作效率和技术水平。具体培训需求如下：

3.20.1 开发人员培训

开发人员接受相关技术和工具的培训，以掌握数据处理、图表生成，前端和后端开发等方面的技术和工具的应用。

3.20.2 测试人员培训

测试人员接受相关测试技能和方法论的培训，以制定合理的测试计划和测试用例，并发现和定位各类问题。

3.20.3 运维人员培训

运维人员接受相关服务器管理和网络管理的培训，以保证系统的高可用性和安全性，并快速定位和解决各类系统问题。

3.20.4 客服人员培训

客服人员接受相关客户服务技能和知识的培训，以回答用户的问题和反馈，并解决用户的问题和需求。

以上是人事管理系统的培训需求，以保证相关人员具备必要的技能和知识，能够高效地完成各自的工作，并提高产品的质量和用户体验。

3.21 有关后勤需求

为保障人事管理系统的正常运营和服务，需要满足以下后勤需求：

3.21.1 办公场地

需要提供足够的办公场地，包括开发团队、测试团队、运维团队和客服团队等不同部门的工作场所。办公场地应符合相关安全、环保和卫生标准，且能够满足团队的办公需求，包括但不限于办公桌椅、电脑、网络等设备。

3.21.2 谈论室（会议室）

需要提供会议室，以满足各部门进行会议和沟通的需求。会议室应配备投影仪、音响设备、白板等工具，以方便会议的进行和记录。

3.21.3 餐饮服务

需要提供餐饮服务，以满足团队成员的工作和生活需求。餐饮服务应健康、卫生、营养丰富，且能够提供多样化的菜品选择，以满足不同口味需求。

3.22 其他需求

3.22.1 可使用性

基于 web 实现人事管理系统，实现用户使用的便捷性。应用可以实现输入信息的存储和处理，并采取多样化形式（表单，图型等）向用户展示查询的内容。

3.22.2 安全性

3.22.2.1 用户隐私

未来保护用户隐私安全，传输用户数据时要使用加密算法，且密码等保密级别极高的数据不在数据库中明文存储。

3.22.2.2 验证系统

由于不同身份的用户对于不同的接口使用权限不同，应有完善的验证系统对不同用户的身份进行检查。保证普通用户，管理员用户可以分别执行对应的操作，确保系统的有序性。

3.22.2.3 SQL 注入

SQL 注入会对后台数据库造成很大的威胁，也增大了用户信息泄露的危险。因此拥有防止 SQL 注入的机制是必要的。该项目通过采用 ORM 等机制来解决 SQL 注入的威胁，保证数据库的安全性。

3.22.2.4 XSS 保护

本项目中有很多需要提交的表单，如果这些地方被植入脚本，很可能对系统造成威胁，所以需要有 XSS 防范机制。

需要使用 XSS 防护机制来对用户可操纵的数据进行 HTML 转义，之后再呈现给用户。如果用户可操纵的内容是 HTML 的标签属性，那么还要进行关键词的黑名单/白名单的过滤。

3.22.3 可维护性

3.22.3.1 完善的日志系统

项目建立实时记录应用行为的运行日志，建立完善的日志系统，当应用出现问题时能够帮助我们发现、解决问题。

3.22.3.2 数据库的可恢复性

需要定期为数据库进行备份，将数据库文件数据备份到另一台服务器。

3.22.3.3 微体系架构

项目的各个功能模块都具有独立性。每个模块的开发运行是独立的，当一个模块运行出问题不会影响当前模块的其他进程，或其余功能模块的正常运行。一些功能应该冗余存储，防止因为个别的节点导致系统瘫痪，进而保证系统运行的稳定性。

3.22.4 可移植性

目为模块化实现。每个模块相对独立，模块之间通过接口进行数据交换。其中的重点功能应该为多个高层的功能提供服务。

3.24 需求的优先次序和关键程度

1. 内容提取的准确性和可靠性，特别是语音转文字功能的精度，在我们的人事管理系统中得到了充分的优化和改进。我们确保录入的员工信息以及其他相关数据都能够准确无误地被提取和存储，以满足用户的需求。

2. 文本摘要的准确性和完整性是我们系统设计的重要考量之一。通过我们系统提供的文本摘要功能,用户可以轻松获取到信息的重点和关键内容,节省了阅读时间,提高了工作效率。
3. 数据安全性和隐私保护是我们系统的核心原则之一。用户上传的音视频和文档都会经过加密处理,并存储在安全的服务器上,以防止数据泄露和未经授权的访问。
4. 用户体验和界面设计的易用性是我们系统设计的重要目标。我们注重用户的操作体验,通过简洁直观的界面设计和用户友好的交互方式,确保用户可以方便地使用和浏览应用程序,提升用户满意度。
5. 思维导图功能在我们的系统中具有高度的可扩展性和灵活性。用户可以根据自己的需求自定义思维导图的结构和内容,使其更符合个性化的工作习惯和需求。
6. 搜索和社区功能的效率和有效性也得到了我们系统的重视和优化。我们通过优化搜索算法和界面设计,确保用户可以快速地找到他们需要的信息和资源,并且能够与其他用户进行有效的交流和互动。

4 合格性规定

本章定义一组合格性方法,对于第 3 章中每个需求,指定所使用的方法,以确保需求得到满足。下面用表格形式表示在第 3 章的每个需求中要使用的方法。

合格性方法包括:

- a. 演示: 运行依赖于可见的功能操作的 CSCI 或部分 CSCI,不需要使用仪器、专用测试设备或进行事后分析;
- b. 测试: 使用仪器或其他专用测试设备运行 CSCI 或部分 CSCI,以便采集数据供事后分析使用;
- c. 分析: 对从其他合格性方法中获得的积累数据进行处理,例如测试结果的归约、解释或推断;
- d. 审查: 对 CSCI 代码、文档等进行可视化检查;
- e. 特殊的合格性方法。任何应用到 CSCI 的特殊合格性方法,如: 专用工具、技术、过程、设施、验收限制。

表 9

需求编号	合格性方法
3.1	演示、测试、审查
3.2	演示、测试、审查
3.3	演示、测试、审查、分析
3.4	演示、测试、审查、分析
3.5	演示、测试、审查、分析
3.6	演示、测试、审查、分析
3.7	演示、测试、审查、分析
3.8	演示、测试、审查、分析

3.9	演示、测试、审查、分析
3.10	演示、测试、审查、分析
3.11	演示、测试、审查、分析
3.12	演示、测试、审查、分析
3.13	演示、测试、审查、分析
3.14	演示、测试、审查、分析
3.15	演示、测试、审查、分析
3.16	演示、测试、审查、分析
3.17	演示、测试、审查、分析
3.18	演示、测试、审查、分析
3.19	演示、测试、审查、分析
3.20	演示、测试、审查、分析
3.21	演示、测试、审查、分析
3.22	演示、测试、审查、分析
3.23	演示、测试、审查、分析
3.24	演示、测试、审查

5 需求可追踪性

本章应包括：

a. 从本规格说明中每个 CSCI 的需求到其所涉及的系统(或子系统)需求的可追踪性。(该可追踪性也可以通过对第 3 章中的每个需求进行注释的方法加以描述)。

注：每一层次的系统细化可能导致对更高层次的需求不能直接进行追踪。例如：建立多个 CSCI 的系统体系结构设计可能会产生有关 CSCI 之间接口的需求，而这些接口需求在系统需求中并没有被覆盖，这样的需求可以被追踪到诸如“系统实现”这样的一般需求，或被追踪到导致它们产生的系统设计决策上。

b. 从分配到被本规格说明中的 CSCI 的每个系统(或子系统)需求到涉及它的 CSCI 需求的可追踪性。分配到 CSCI 的所有系统(或子系统)需求应加以说明。追踪到 IRS 中所包含的 CSCI 需求可引用 IRS。

5.1 注册、登录功能

5.2 查询工资、考核功能

5.3 修改个人信息功能

5.1 查询用户信息，工资，考核

- 系统需求：能够从数据库中准确反馈出员工基本信息，用户的个人信息，工资发放记录，违规、绩效记录。
- CSCI 需求：能够整合计算上一个月，本年度该员工的绩效，考核成绩，工资情况。

5.2 管理用户

- 系统需求：管理员用户能够查询所有员工相关信息；登记用户工作过程中违规、考核记录；升降员工职位，工资；聘用、解聘员工。
- CSCI 需求：根据工作年限，项目完成情况，员工绩效，工资评估所有员工的工作质量排名，并生成图表等。

5.3 员工修改信息功能

- 系统需求：员工可以修改自己的个人信息（除绩效，违规，工资等）。
- CSCI 需求：员工申请调岗，加薪，离职或修改普通个人信息等。

5.4 提交申请

- 系统需求：普通用户可以通过系统提交加班、请假、调岗等申请，填写申请表单并提交审批。
- CSCI 需求：用户可以选择申请类型、填写申请原因和期间，并上传相关附件。

6 尚未解决的问题

如需要，可说明软件需求中的尚未解决的遗留问题。

- 如何提高人事管理系统中员工信息提取的准确率？
- 如何避免人事管理系统中员工信息上传时存在敏感信息的风险？
- 人事管理系统是否需要支持多种语言的界面和文档处理功能？
- 人事管理系统中社区功能（如内部论坛）的具体实现方式和安全性如何保障？
- 如何优化人事管理系统中员工信息搜索功能的效率和准确性？
- 如何解决人事管理系统因网络不稳定或者处理大数据量导致的系统崩溃问题？

7 注解

本章应包含有助于理解本文档的一般信息(例如背景信息、词汇表、原理)。本章应包含为理解本文档需要的术语和定义，所有缩略语和它们在文档中的含义的字母序列表。

7.1 术语和定义

本文档涉及的一些术语和定义如下：

1. 功能性需求：系统需求之一切方面，如功能性需求、性能需求、接口需求、设计约束、质量属性等。
2. 非功能性需求：除功能性需求之外的其他需求，如保密性、私密性、数据管理等。
3. 输入：该功能的所有输入数据，包括输入源、数量、度量单位、时间设定和有效输入范围等。
4. 输出：该功能的预期输出结果，包括输出数据的有效性检查、数据元素集合体中的数据元素及其结构等。
5. 数据元素集合体：包含记录、消息、文件、显示和报表等数据的集合体。
6. 保密性和私密性：对输入输出、用户界面设计、数据管理和故障处理等方面的需求做了详细描述。
7. 基线：在软件设计中，基线是指项目生命周期中特定的工作产品被正式批准，并作为后续工作的基准，任何更改都需要经过正式的变更控制过程。基线有助于管理软件变更，确保项目团队成员对项目状态有共同的理解，并提供一个恢复点以防后续变更导致问题。
8. LDAP（轻量级目录访问协议，Lightweight Directory Access Protocol）和 AD（活动目录，Active Directory）是用于管理网络中用户、计算机、打印机等资源的目录服务协议和系统。

7.2 词汇表

1. 软件需求规格说明（SRS）：为开发团队、客户以及其他相关人员提供详细的系统开发需求规格说明的文档。
2. 2 CSCI：表示“计算机软件配置项”，即计算机软件的一部分。
3. IRS：表示“软件需求规格说明书”，是对软件系统所需的功能、性能和约束条件进行详细说明了的文档。
4. 可追踪性：指需求或设计元素与软件或系统的其他元素之间的联系，以便能够跟踪软件或系统的不同部分之间的关系。

附录

附录可用来提供那些为便于文档维护而单独出版的信息(例如图表、分类数据)。为便于处理，附录可单独装订成册。附录应按字母顺序(A，B 等)编排。